tu/e
Research Report
ISSN 0167-9708
Coden: TEUEDE

Eindhoven
University of Technology
Netherlands

Faculty of Electrical Engineering

LASER: A LAyout Sensitivity ExploreR Report and User's Manual

by
Jose Pineda de Gyvez

EUT Report 89-E-216
ISBN 90-6144-216-8

March 1989

Eindhoven University of Technology Research Reports.
EINDHOVEN UNIVERSITY OF TECHNOLOGY

Faculty of Electrical Engineering
Eindhoven The Netherlands

ISSN 0167-9708 Coden:TEUEDE

LASER: A LAyout Sensitivity ExploreR
Report and User's Manual

by
Jose Pineda de Gyvez

EUT Report 89-E-216
ISBN 90-6144-216-8

Eindhoven
March 1989

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Pineda de Gyvez, J.

LASER: a layout sensitivity explorer. Report and user's manual /
by J. Pineda de Gyvez. - Eindhoven: University of Technology,
Faculty of Electrical Engineering. - Fig. - (EUT report,
ISSN 0167-9708; 89-E-216)
Met lit. opg., reg.
ISBN 90-6144-216-8
SISO 663.42 UDC 621.382:681.3.06 NUGI 832
Trefw.: elektronische schakelingen; computer aided design.

Abstract

As the IC pattern resolutions tend to become smaller the layout geometry plays a more important role in IC yield. The probability that a chip will fail is directly related to the way that the IC artwork is laid out. By examining the possible places where catastrophic defects may occur one can prevent potential faults, and thus estimate the reliability of the design. Rrealistic yield simulation tools must consider the specific layout. It is, therefore, ideal a CAE tool that automatically explores and predicts the layout reliability for real environmental conditions prevailing in the manufacturing line. We present a system capable of interactively finding the critical areas for shorts and breaks, the sensitivity, and the yield of the IC artwork, for any range of defect sizes. The implementation is based on a simple scanline algorithm and performs only one layout extraction for any span of defect sizes

Pineda de Gyvez, J
LASER: A LAyout Sensitivity ExploreR. Report and User's manual
Faculty of Electrical Engineering, Eindhoven University of Technology, 1989.
EUT Report 89-E-216

Author's address:
Automatic System Design Group,
Faculty of Electrical Engineering,
Eindhoven University of Technology,
P.O. Box 513
5600 MB Eindhoven,
The Netherlands

CONTENTS

1. INTRODUCTION. 1

2. SENSITIVE AREAS. 2
   2.1 PATTERN SENSITIVE AREA FOR BRIDGES 2
   2.2 PATTERN SENSITIVE AREA FOR CUTS 3

3. FINDING THE LAYOUT CRITICAL AREAS. 5

4. EXTRACTION OF SUSCEPTIBLE SITES. 7

5. CREATION AND COMPUTATION OF CRITICAL AREAS. 10

6. SENSITIVITY, PROBABILITY OF FAILURE AND YIELD. 14

7. LAYOUT ANALYSIS. 16

8. CONCLUSIONS 27

9. REFERENCES 28

APPENDIX: User's Manual

LIST OF FIGURES

Figure 1. Pattern sensitive area for shorts 3
Figure 2. Pattern sensitive area for breaks 3
Figure 3. Two different critical areas. (a) a single conductor. (b) Critical area if the current is applied at the left or right side of the pattern. (c) Critical area when the current is applied at the top or bottom of the pattern. 4
Figure 4. Forming the critical areas geometrically by modifying the edges of the patterns.(a)Bridges. (b)Cuts. 5
Figure 5. The critical areas for both bridges and cuts are found by shrinking the susceptibles sites 6
Figure 6. Susceptible sites for bridges 7
Figure 7. Susceptible sites for cuts. (a) The sender terminals are on the top and lower right end. (b) The sender terminals are on the bottom and left side. 7
Figure 8. Creation of susceptible sites for bridges 9
Figure 9. Creating critical regions for bridges from their susceptible sites. (a) Susceptible sites from the vertical sweep. (b) Susceptible sites from the horizontal sweep. (c) Critical regions formed. 10
Figure 10. Creating critical regions for cuts from their susceptible sites. (a) Susceptible sites from the vertical sweep. (b) Susceptible sites from the horizontal sweep. (c) Critical regions formed. 11
Figure 12. Overlapping regions in the computation of critical areas 11
Figure 11. Creation of critical regions 12
Figure 13. Computation of the critical areas 13
Figure 14. (a)PLA. (b)TM. (c)STD 17
Figure 15. Critical regions are shown in black.(a)Bridges (b)Cuts. 18
Figure 16. Layout Sensitivity for bridges 19
Figure 17. Layout sensitivity to cuts 21
Figure 18. Defect Size Distribution 22
Figure 19. Layout Probability of Failure (POF) for bridges 23
Figure 20. Layout Probability of failure (POF) for cuts 24
Figure 21. Mask yield 25
Figure 22. Final Layout yield 26

LASER

Report

1. INTRODUCTION.

Traditional layout verification is the task of validating the design rules imposed by the technological process, i.e. verifying the width of patterns, the space between them, etc. Under this approach, external sources that can lead to incorrect layouts are in most of the cases not taken into account. These external sources manifest themselves in the form of "layout contaminants", i.e. undesired dust particles that drastically change the shape of a pattern, and as a result the affected region is said to be a "defect" in the layout. Nowadays the significance of these defects is crucial to the successful manufacturing of the chip in spite of a precise control of the line features.

As the processes tend to mature and to advance to smaller resolution features other forms of layout verfication become imperative. One such form of verification is to foresee the artwork endurance in real manufacturing environments and a mean to achieve this is by finding the critical areas where a short or a break can happen in the layout. Critical areas have a lot of potential due to the ease of predicting the IC artwork yield [1]. Evidence exists [2] that there are programs for computing them, however, they have not been reported in the literature. Also due to the complexity of the layouts, several authors prefer to derive equivalent layouts to simplify the problem, as is the case of the concept of "virtual layouts" introduced in [7] and the one of "equivalent layouts" in [8].

It is desirable that a layout sensitivity extractor finds the critical areas for several defect sizes at the time. Otherwise the system can be impractical from the efficiency point of view since each extraction costs the designer's time not to mention the computational resources involved. Moreover, one is usually interested in a range of defect sizes and not in solely one.

Up to now it was possible to find the critical areas for complex layouts using a statistical Monte Carlo simulation, and, analytical methodologies were restricted only to very regular and simple layouts. We present a layout verifier capable of identifying the critical areas in complex layouts. Unlikely to Monte Carlo approaches our methodology is based purely on the geometry of the patterns. The implementation is based on a scanline algorithm and performs only one layout extraction for any span of defect sizes.

The extraction of critical areas forms part of a large system for IC layout yield estimation. The system has capabilities to display the critical regions onscreen, and provides facilities such as: the computation of the layout sensitivity, a flexible defect size distribution that can easily be fit to real manufacturing conditions, the computation of the layout probability of failure for a given defect size distribution, and the prediction of the artwork yield for any span of defect sizes.

2. SENSITIVE AREAS.

It is convenient to examine briefly the nature of defects and their impact in the ICs. An IC layer is a piece of solid state surface in the wafer, usually shaped by one or more masks. Each pattern in the layer has a set of properties such as the mask-layer shape matching, the integrity of the pattern, its thickness in the layer, etc. Non desirable physical agents introduce changes in the structure of a group of patterns and as a result the properties of these patterns are different from the initial ones. These changes are known as defects. For our purposes, a defect is any deviation in the shape of the IC layer from its corresponding layout mask. For instance the absence of a piece of mask can represent the absence or presence of a certain spot of material in a specific layer. Defects can introduce faults. A fault is any deviation from the expected behaviour of the IC. Some faults are fatal, such as stuck-at outputs or dc-path changes in the topology of the circuit, other forms are just performance failures like undesirable delays. In our study we concentrate on spot defects that cause catastrophic faults[17].

The layout is the union of geometrical structures with an electrical meaning, i.e. a via, a transistor, a wire, etc. Depending upon the structure, the patterns in each mask have a significance other than simple conductores, i.e. a poly pattern over a diffusion pattern form a transistor. Therefore we distinguish two kinds of critical areas [6]: 1) "Pattern Sensitive Area" is the area where the center of a defect must fall to cause a fault to the pattern, such as breaking it or joining it with another pattern. 2) "Structure Sensitive Area", on the other hand, is the area where the center of a defect must be situated in order to introduce a fault to a complete electrical structure, like a transistor, a via, etc.

For layout verification purposes the first kind of sensitive areas are of interest. Notice that these critical areas are more concerned with the layout rather than with the electrical circuit. The fault models considered are only two, namely:

— the *bridge* (undesired joined patterns)

— the *cut* (undesired broken patterns)

In our approach we find the critical areas per mask, thus no interdependance between masks is considered. It is difficult to model the shape of defects since in reality they are rough-edged splotches. Hence, modeling defects as squares provides a solution that is sufficiently correct, besides that the algorithms become much faster and simpler.

2.1 PATTERN SENSITIVE AREA FOR BRIDGES

Under the pattern approach all the patterns are considered as interconnectors and a fault appears only when nonequipotential regions are joined together. Fig. 1 shows the case of two single conductive lines, each of width w, length L and space s between them, with $L \gg s$. Assume that an extra spot of material occurs between the two conductors, if the size of this defect is such that $x > s$ the critical area can be expressed as:

$$A_{PS} = W_{PS} L_{PS} \tag{1}$$

where

$W_{PS} = x - s$
$L_{PS} = L + x$

W_{PS} and L_{PS} represent the critical width and length of the critical area, respectively. The end effects of the defect are also accounted in the length of the critical area.

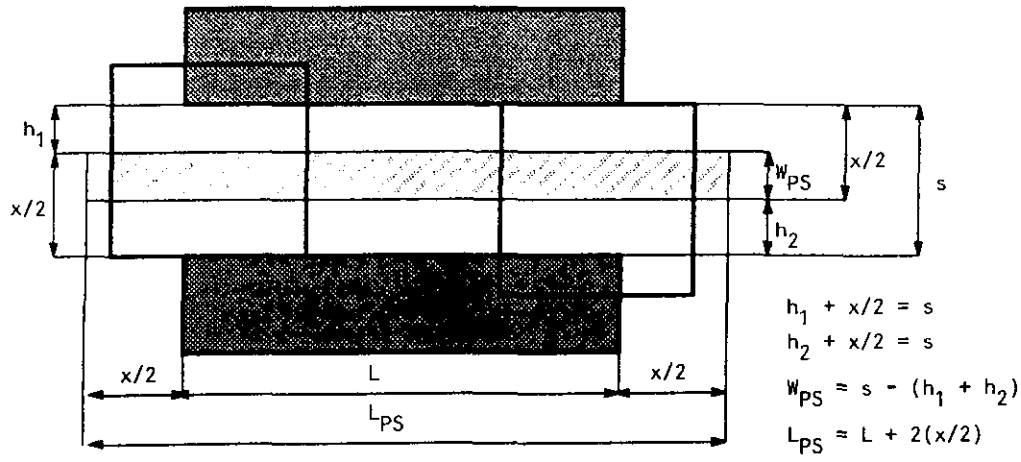


Figure 1. Pattern sensitive area for shorts

2.2 PATTERN SENSITIVE AREA FOR CUTS

A cut appears when an equipotential region is fragmented into two or more nonequipotential patterns. Fig. 2 represents a single conductor of widht w and length L, with $L \gg w$. Assume that a defect in the form of missing material occurs. If the size is such that $x > w$ the critical area can be modelled as:

$$A_{PB} = W_{PB} L_{PB} \tag{2}$$

where

$W_{PB} = x - w$
$L_{PB} = L + x$

W_{PB} and L_{PB} represent the width and length of the critical area respectively. The end effects are also consider in the length of the critical area.

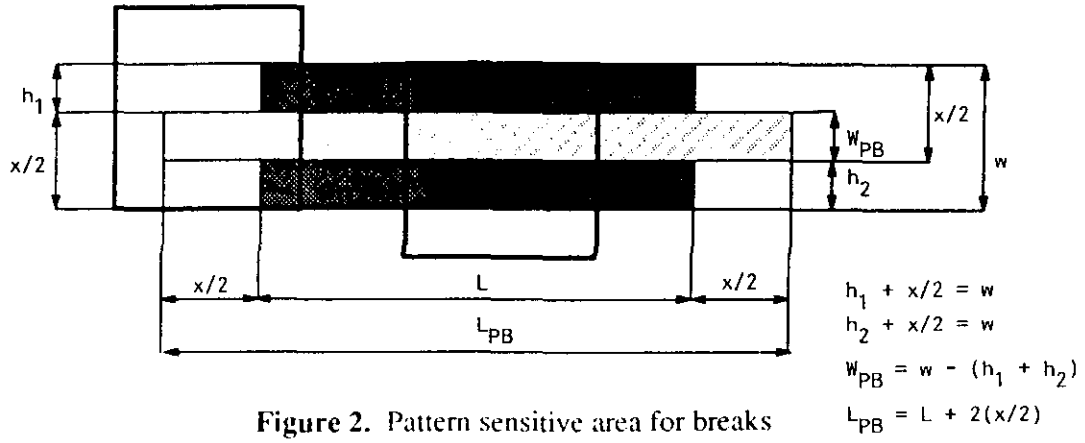


Figure 2. Pattern sensitive area for breaks

The problem of dealing with lose patterns is that the flow of the current must be known in order to decide if the pattern is fragmented in nonequipotential regions. A brief example helps to visualize this. Assume that we have a pattern of length a and width $2a$, as shown in Fig. 3a. If the current is injected at the left side of the pattern and received at the right side, then a defect of size x, such that $a < x < 2a$, will not introduce a fault even if it completely cuts the pattern in the vertical direction. The reason why is because there is still continuity between the terminals where the current is injected and where it is

expected to arrive. In this case only defects of size $x>2a$ can introduce a fault, and, the critical areas would be in the horizontal direction as depictured in Fig. 3b. Assume now that the current is injected at the top and received at the bottom of the pattern. For this situation defects of size $x>a$ are fatal because the pattern can be broken in two nonequipotential regions if the defect is situated anywhere along the critical area depictured in Fig. 3c.

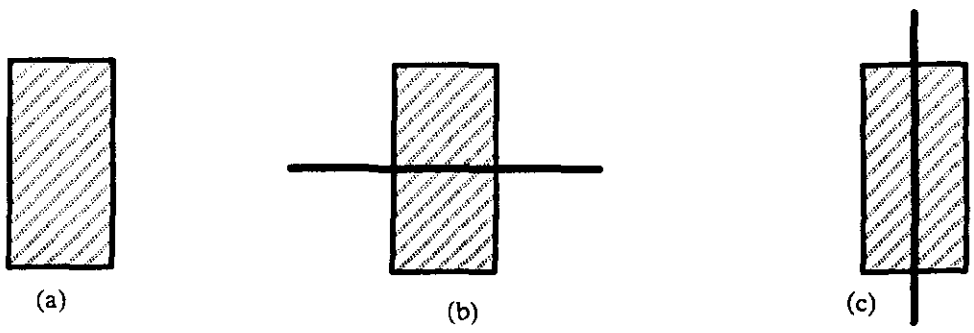


Figure 3. Two different critical areas. (a) a single conductor. (b) Critical area if the current is applied at the left or right side of the pattern. (c) Critical area when the current is applied at the top or bottom of the pattern.

We denote the terminals where the current is applied as *senders* and the terminals that the current has to reach as *receivers*. These terminals play an important role in our solution to the problem of finding the critical areas, as it will be seen later.

3. FINDING THE LAYOUT CRITICAL AREAS.

A great deal of computational efforts can be saved if the critical areas are found geometrically rather than analytically. Under the geometrical approach [12,13], the critical areas for bridges are found by expanding each pattern by an amount equal to half of the defect size, and then by comparing if the expansions intersect. If so, then the amount of intersection corresponds to the critical area between the patterns, see Fig. 4a. In the case of critical areas for cuts, every pattern is shrinked by half of the defect size, and a critical area is established only when the shrinked edges intersect each other, see Fig. 4b.

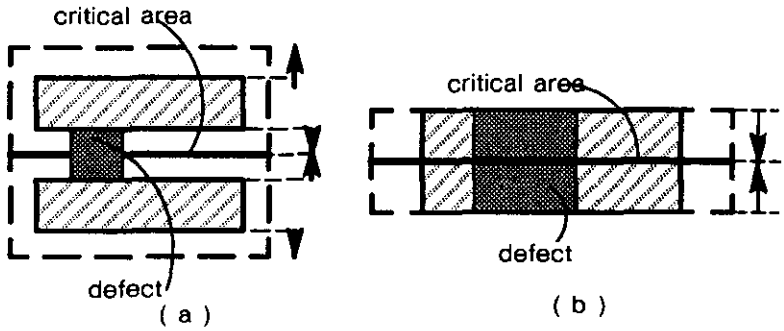


Figure 4. Forming the critical areas geometrically by modifying the edges of the patterns.(a)Bridges. (b)Cuts.

The simplest way to extract the critical areas for bridges is to compare a pattern against all the other patterns but this is computationally prohibitive. Also, the expansion of the patterns implies that a layout extraction has to be executed for each defect size. For the areas sensitive to cuts the problem is simpler since no neighbour patterns are necessary. Hence the problem is reduced to shrink the patterns and to search for intersections among them, like corners or overlaps.

Our algorithm is based on the geometrical approach explained above. However, instead of proceeding directly to establish the critical areas we determine first the regions where a defect of any size can cause a fault. Namely, these regions are the empty spaces between patterns, for the case of bridges, and the patterns themselves, for the case of cuts. This results in a single layout extraction for any defect size. Furthermore, for bridges it is no longer necessary to compare the patterns among each other since the places where defects can cause faults are the empty spaces. Therefore the problem is reduced to determine the extent of the critical area from these regions according to the defect size.

For terminology purposes, we denote the former regions as *susceptible sites for bridges*, and the latter ones as *susceptible sites for cuts*.

We outline now the steps involved in the extraction of critical areas and later we describe them into more detail. These steps are mainly a layout extraction to obtain the susceptible sites, the creation of critical regions from the susceptible sites, and the computation of the critical areas themselves.

Step 0. Select the layout mask to be analysed. Convert the rectangles to polygons. Assign unique identification numbers to the polygons. Decompose the polygons to line segments. Save these line segments for further processing.

Step 1. Extract all the susceptible sites for bridges and cuts from the pre-processed mask and store them in two different data structures, one for bridges and one for cuts. We denote these data structures as "susceptible structures".

Step 2. For every defect size traverse the "susceptible" structure in order to obtain the coordinates of the corresponding critical area. These coordinates delineate the boundary of the critical area. The coordinates of areas sensitive to bridges are obtained by shrinking the related susceptible sites as if we were dealing with a cut except that in this case the coordinates belong in fact to the critical area for a bridge between the patterns, see Fig. 5. The areas sensitive to cuts are found using the same procedure. Store the coordinates in two independent data structures, one for bridges and another for cuts, each one of them indexed by its defect size. Repeat this step until all the range of defect sizes is exhausted.

Step 3. For every defect size compute the total critical area enclosed in the set of coordinates found above. The total critical area per defect size is the union of the areas of the individual boundaries found in step 2. Notice, that the critical areas can overlap, hence the total critical area is the union and not the sum of the area of each boundary.

From the procedure above it can be seen that only one layout extraction is necessary (steps 0,1) and also that no matter how large the defect size is the identification of critical areas will always be found in a linear time proportional to the number of susceptible sites, which in turn reflect the complexity of the layout (step 2).

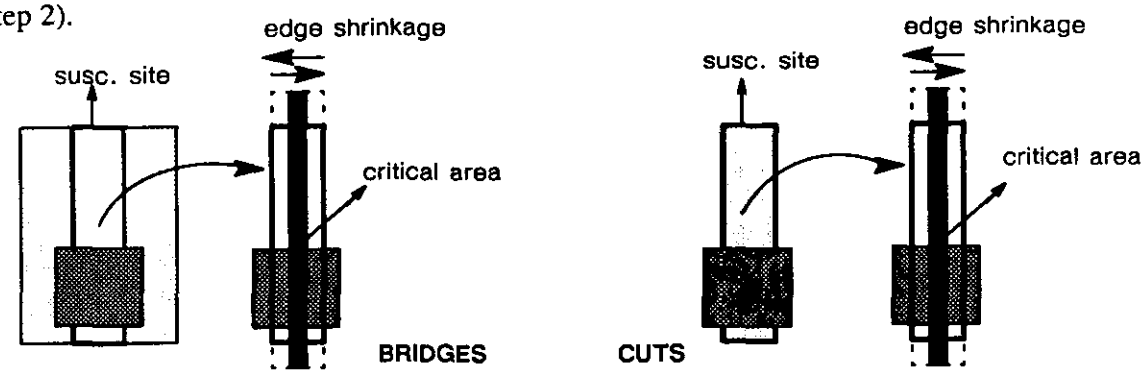


Figure 5. The critical areas for both bridges and cuts are found by shrinking the susceptibles sites

4. EXTRACTION OF SUSCEPTIBLE SITES.

The susceptible sites for bridges contain all the necessary information to find the related critical areas. That is, their boundaries represent in fact the edges of the associated patterns which can be joined by a defect. Thus, it is no longer necessary to compare patterns in order to establish the critical area, and neither is necessary to extract the layout for every distinct defect size. Fig. 6 shows an example of these sites.

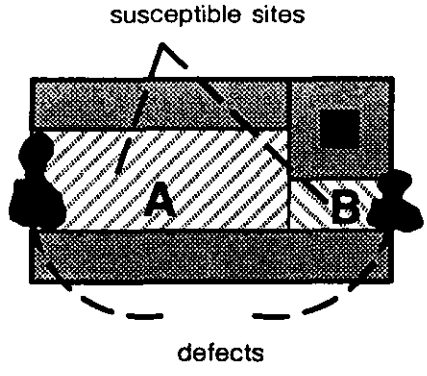


Figure 6. Susceptible sites for bridges

The susceptible sites for cuts provide the correct magnitude of the pattern from which the critical area can be found straightforwardly. For instance, a polygon with an "L" shape contains four different susceptible sites. If the sender terminals are on the top and lower right end of the "L", the susceptible sites are as depictured in Fig. 7a. If, on the other hand, the sender terminals are situated on the bottom and left side then the susceptible sites are the ones shown in Fig. 7b.

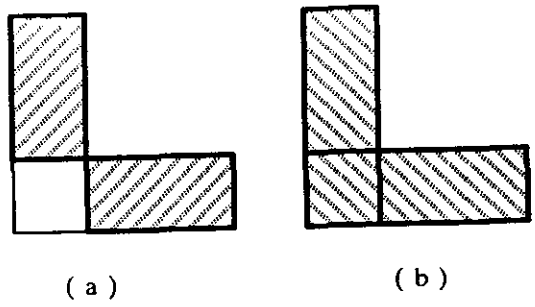


Figure 7. Susceptible sites for cuts. (a) The sender terminals are on the top and lower right end. (b) The sender terminals are on the bottom and left side.

Before extracting the susceptible sites we perform a layout preprocessing. It consists of converting rectangles to polygons, and of assigning unique identification numbers to every polygon. For subsequent processing the boundary of the polygons is decomposed in line segments. Each line segment inherits the identification number of the polygon. We further classify the line segments in two types. The vertical (horizontal) line segments at the bottom (left) of the polygon are of type BEGIN, and the ones at the top (right) are of type END. Each vertical line segment is specified by its x-coordinate and the y-values of the lower and upper endpoints. Each horizontal line segment is similarly specified by its y-coordinate and the x-values of its left and right endpoints. Finally, we store the horizontal and vertical line segments in two different data structures.

We use a scanline algorithm [10] to extract the susceptible sites. In each case we perform two opposite layout sweeps: A bottom-up sweep that covers all the susceptible sites parallel to the scanline, and a left-right sweep that encompasses all the sites parallel to this scanline. The top-down sweep, or VERTICAL sweep, scans the horizontal line segments. The left-right sweep, or HORIZONTAL sweep, the vertical ones. As the algorithms for finding susceptible sites for bridges and cuts are very similar the explanation to follow is restricted to bridges. Later we mention the changes to mend the same algorithm for cuts.

In each sweep a susceptible site is identified when a BEGIN line segment and an END line segment are parallel and adjacent to each other. The susceptible sites have rectangular shape where the width is defined by the intersection of the BEGIN and END line segments, and the length by the distance between the two line segments. To cover possible bridges at the corners, we also find the line segments immediately to the left and immediately to the right of the BEGIN line segment. When a susceptible site is formed we label it as VERTICAL, or HORIZONTAL, depending on the sweep in which it was found. An exception to this labeling is when the susceptible site was formed from a corner of the BEGIN line segment. In this case we label the susceptible site as CORNER. The justification of labeling the susceptible sites is because in step 2 we need to know what coordinates to shrink, either the abscissae or the ordinates before we can deduce if a critical area is established. Since the critical areas tend to grow as the defect size grows and thus to overlap each other, there is no need to look for more neighbour line segments to form the susceptible sites. This effect of overlapping of critical areas is known as the "proximity effect" [2].

Let $P = \{ p_{i\,1}, \cdots, p_{i\,n} \}$ be the set of polygon line segments in ascending y-order. M the order relation of line segments sorted in ascending x-order, and S a data structure to store the susceptible sites found. The main loop of the algorithm sweeps a scanline through the set P. We use the data structure M to store the END line segments ordered by their left-coordinates. Initially M is empty. Whenever a BEGIN line segment is encountered during the sweep we check for intersections with the horizontal line segments in M. We also look, in M, for the line segment immediately to the left (predecessor) and to the right (successor) of the BEGIN line segment. The data structure M is updated in such a form that only the intersected sections of the intersected line segment are deleted from M. If an intersection was found, a new susceptible site is made and stored in the S data structure. The algorithm is directly applicable for the vertical sweep. For the horizontal sweep the coordinates have to be rotated 90 degrees clockwise. That is, the lower coordinate of the vertical line segment becomes the left coordinate, the top coordinate becomes the right one, and the x-value becomes the y-value. Whenever a susceptible site is formed in this sweep the coordinates must be rotated 90 degrees counterclockwise in order to set the correct orientation of the site. The algorithm for finding susceptible sites related to bridges is shown in Fig. 8.

MAKE_SITE (*left* , *right* , *bottom* , *top*) creates a susceptible site with the given coordinates

```
M ← ∅
S ← ∅
predecessor = ∅
successor = ∅

FOREACH ( p_i ∈ P ) {                          /* for every edge of the polygon    */
  IF ( p_i .type = END )                       /* if the edge type is END, insert  */
   M ← p_i                                     /* the edge in the M data structure */
  ELSE {
    FOREACH ( m_i ∈ M ) {                      /* for every edge in M */
      IF ( m_i . right < p_i . left )          /* determine the predecessor edge of */
          predecessor = m_i                    /* the polygon edge                  */
      IF ( m_i . left > p_i . right ) {        /* determine the successor edge, and */
       successor = m_i                         /* exit the search                   */
        BREAK
      }                                        /* if the polygon edge intersects    */
                                               /* the stored edge                   */
      IF ( ( p_i ∩ m_i ) ≠ ∅ ) {
                                               /* if the id's of the edges are
                                               /* different, make a susceptible site and */
                                               /* reinstall the nonintersected      */
                                               /* sections of the retrieved edge    */
        IF ( p_i .id ≠ m_i .id ) {
         S ← MAKE_SITE (MAX (p. left , m_i . left ),MIN ( p_i . right ,m. right ),
                        m_i . top , p_i . bottom ,sweep)
         M ← m_i - ( m_i ∩ p_i )
        }
      }
    }                                          /* make possible susceptible sites at */
                                               /*  the extremes of the polygon edge  */
    IF ( predecessor ≠ ∅ )
     S ← MAKE_SITE ( predecessor. right , p_i . left ,
                     predecessor. top , p_i . bottom ,CORNER)
    IF ( successor ≠ ∅ )
     S ← MAKE_SITE ( p_i . right ,successor. left ,
                     successor. top , p_i . bottom ,CORNER)
  }
}
```

Figure 8. Creation of susceptible sites for bridges

The extraction of susceptible sites for breaks is essentially the same except that in the algorithm the END line segments are the ones that are stored in the data structure M, and instead of processing line segments of different id's the id's must be the same. In some situations the algorithm will apparently find the same critical region for both vertical and horizontal sweeps. This is because we assumed that the "sender terminals" can be either vertical or horizontal. Recall from our example of Fig. 3 that a simple conductor has two different critical regions and only one is in effect depending upon where the "sender terminals" are situated. Since we are dealing with independent masks there is no way to know the signal flow (left to right, top to bottom ?). That is why the both critical regions are created.

5. CREATION AND COMPUTATION OF CRITICAL AREAS.

The task concerning the computation of critical areas is divided in two phases. Namely, creating the boundary of the critical area for each defect size, and then computing the area enclosed in the boundary. To simplify the terminology in the context of this paper we name the boundary of the critical area as *critical region*, and the area enclosed in the boundary as *critical area*.

As we mentioned before, the susceptible sites are the potential places where a defect of any size can cause a fault. The critical regions related to bridges are found by traversing the correponding susceptible data structure. The edges of each susceptible site are shrinked by half of the defect size to test if the critical region is established. When the site is labeled as: 1)VERTICAL the abscissae are shrinked, 2)HORIZONTAL the ordinates are shrinked, and 3)CORNER both the ordinates and abscissae are shrinked. In any case, if the shrinked edges overlap then the critical region is established. The areas sensitive to cuts are found exactly in the same manner, except that the data structure traversed is the one related to cuts.

An example will help to visualize the creation of critical areas. Let us first consider the case of critical areas for bridges. Assume that we have two L shaped conductors, running parallel to each other, with space s between them and that a defect of size $s<x<L$ is placed among them. The susceptible sites for bridges are identified as A and B. Susceptible site A was obtained in the vertical sweep, see Fig. 9a, thus their abscissae are shrinked. Susceptible site B was obtained in the horizontal sweep, see Fig. 9b, hence the ordinates are shrinked. The resulting critical regions are shown in Fig. 9c.

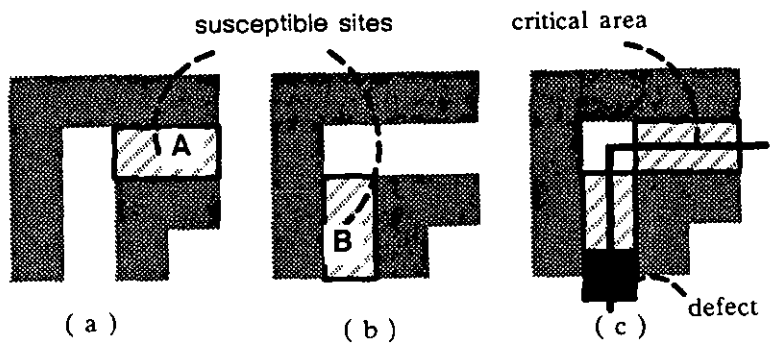


Figure 9. Creating critical regions for bridges from their susceptible sites. (a) Susceptible sites from the vertical sweep. (b) Susceptible sites from the horizontal sweep. (c) Critical regions formed.

Now let us consider the case of cuts for a conductor with also an L shape. Fig. 10a shows the susceptible sites A and B which were extracted during the vertical sweep. Fig. 10b shows the suceptible sites C and D extracted during the horizontal sweep. For the former sites the abscissae are shrinked and for the latter sites the ordinates. Fig. 10c shows the critical areas for a defect of size $w<x<L$. Notice that no critical area was established for the susceptible sites A and D. The reason is because the shrinked edges for these sites do not overlap for the defect size.

If the defect size is of such a magnitude that it joints or cuts more patterns than the ones related in the susceptible sites, the critical regions will still be found correctly because the shrinked edges overlap also. This situation was previously mentioned as the proximity effect.

From this example it is seen that the critical regions are found straightforwardly from the susceptible sites for any defect size. Thus, no matter how large the defect size is the critical regions are extracted in

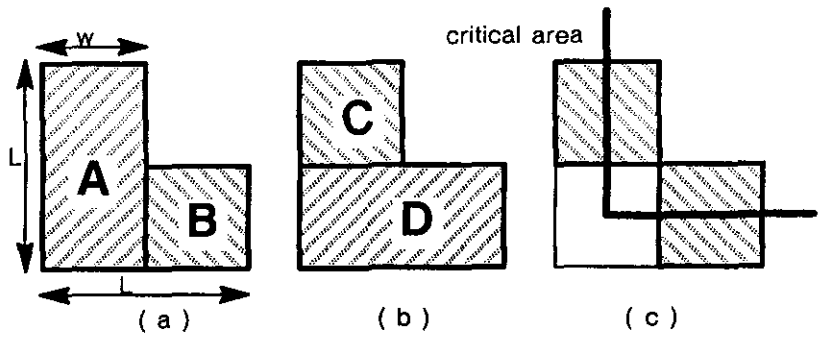
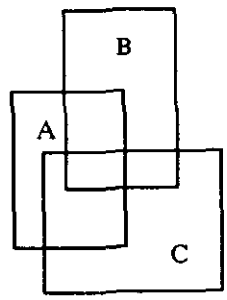


Figure 10. Creating critical regions for cuts from their susceptible sites. (a) Susceptible sites from the vertical sweep. (b) Susceptible sites from the horizontal sweep. (c) Critical regions formed.

a time proportional to the number of susceptible sites.

The algorithm developed to find the critical regions is the same for bridges and for cuts. However the data structures used to store the critical regions must be independent, one for each type of fault. Let $S = \{s_1, \cdots, s_n\}$ be the set of the susceptible sites in ascending y-order. The main loop of the algorithm traverses the set S. Foreach susceptible site we check its label and then see if the critical region, for a given defect size, is established by shrinking the appropriate edges. If a critical region is formed we saved it in the data structure C indexed by the defect size. This procedure is repeated until the range of defect sizes is exhausted. Worthnoticing is that each defect size has its unique critical regions in the layout, and that in fact the critical regions compose a "critical mask" specifically for the defect size in turn. The algorithm is shown in Fig. 11 (see page 12).

It would have been nice if it were possible to compute the critical area immediately after every "critical region" was found just by finding the area of the rectangle. However this is not possible because the critical areas tend to overlap and it is necessary to compensate the total critical area by subtracting the area of even number of overlaps and by adding the area of odd number of overlaps, see Fig 12. This implies book-keeping of overlaps and every time search for intersections. These kind of problems belong to the area of computational geometry and for a vast number of examples the reader is referred to [11]. A simple way to overcome the bookkeping of intersections is to use once more a scanline algorithm.



$$\text{Area} = A + B + C - A \cap B - A \cap C - B \cap C + A \cap B \cap C$$

Figure 12. Overlapping regions in the computation of critical areas

The algorithm runs a scanline from the bottom to the top of the critical mask, and computes the area of all the rectangle's sections that lie between the previous scanline position and the current position. The

```
FOREACH ( defect_size ) {                    /* for each defect size ... */
 displace = defect_size/2                    /* calculate the magnitude that the edges */
  FOREACH ( s_i ∈ S ) {                      /* of the susceptible sites have to be shrinked */
   SWITCH ( s_i .label ) {                   /* depending upon in which sweep was */
   HORIZONTAL:                               /* the susceptible site found take an action */
     c. left = s_i . right - displace        /* shrink the abscissae */
     c. right = s_i . left + displace
     if ( c. left ≤ c. right ) {             /* if they intersect form a */
       c. bottom = s_i . bottom - displace   /* critical region */
       c. top = s_i . top + displace
       C[defect_size]←c
     }
     BREAK
   VERTICAL:
     c. bottom = s_i . top - displace        /* shrink the ordinates */
     c. top = s_i . bottom + displace
     if ( c. bottom ≤ c. top ) {             /* if they intersect form a */
       c. left = s_i . left - displace       /* critical region */
       c. right = s_i . right + displace
       C[defect_size]←c
     }
     BREAK
   CORNER:
     c. left = s_i . right - displace        /* shrink the abscissae and */
     c. right = s_i . left + displace        /* the ordinates */
     c. bottom = s_i . top - displace        /* if the abscissae intersect */
     c. top = s_i . bottom + displace        /* each other, and the ordinates also do */
     if ( c. left ≤ c. right ∧ c. bottom ≤ c. top )
       C[defect_size]←c                      /* form a critical region */
     BREAK
   }
  }
}
```

Figure 11. Creation of critical regions

scan position is updated everytime that the top of a rectangle is found. The scanning stops when the top of the critical mask is reached. We denote the bottom of a critical region as V_BEGIN and the top as V_END. Let $C = \{c_1, \cdots, c_n\}$ be the set of line segments of the critical regions in ascending y-order. M the order relation of line segments sorted in ascending x-order. The main loop of the algorithm sweeps a scanline through the set C. We store the V_BEGIN line segments twice in the M data structure. The first time ordered by their left coordinate, denoted as H_BEGIN, and the second time ordered by their right coordinate, denoted as H_END. Initially M is empty. Whenever a V_END line segment is found during the sweep we check if it is above the last scan position. If so, we proceed to sweep the M data structure. We determine the width of the slice of area from every pair of line segments in M. The length from the current y-value of the V_END line segment and the y-value of the first horizontal line segment retrieved, or the y-value of the last scan position, whichever is applicable. An auxiliary variable helps to discover if two or more rectangles overlap. Everytime that a H_BEGIN (H_END) line segment comes the variable is incremented (decremented). Thus, when the value of the variable is zero it means that a new nonintersected rectangle comes. Every line segment retrieved from M is deleted to update the data structure. The algorithm is presented in Fig. 13.

```
old_top = - ∞
top = - ∞
area = 0

FOREACH ( c_i ∈ C[defect_size] ) {          /* for every edge of the "critical rectangles ... */
  IF ( c_i .type = BEGIN ) {                /* if the edge type is BEGIN insert it twice */
   M← c_i ( left )                          /* in the M data structure */
   M← c_i ( right )
  }
  ELSE {
    IF ( c_i . top >top ) {                 /* if the current edge is above the scan position ... */
     top = c_i . top                        /* update the scan position */
     nest = 0
      FOREACH ( m_i ∈M ) {                  /* for every horizontal edge */
        IF ( nest = 0 ) {                   /* if it is not intersected by other horizontal edge*/
          bottom = ∞                        /* reset the width and length */
          length = width = 0
        }
        ELSE {                              /* calculate the new width */
          width = m_i .key - last_key       /* and length */
          length = top - bottom
          area = area + length*width        /* update the total area */
        }
       last_key = m_i .key                  /* safe the last horizontal position */
       bottom = MIN ( m_i . bottom ,bottom )        /* update the bottom and */
        IF ( bottom<old_top )               /* clip it if necessary */
          bottom = old_top
        IF ( m_i .type = BEGIN )            /* if the horizontal edge is */
          nest = nest + 1                   /*BEGIN increment the nesting*/
        ELSE                                /*level,else,decrement it*/
          nest = nest - 1
      }
     old_top = top                          /*update the last scan position*/
    }
   M = M - m_i                              /*delete the horizontal edge*/
  }
}
```

Figure 13. Computation of the critical areas

6. SENSITIVITY, PROBABILITY OF FAILURE AND YIELD.

In order to predict the layout yield it is necessary to study the impact that the manufacturing conditions will have on the layout. These analyses can be carried on analytically since the critical areas of the layout can be extracted for a span of defect sizes as we showed in previous sections, and thus they are known. The task of layout endurance analysis can be very fast and furthermore there is no need to recur to Monte Carlo statistical simulations to obtain the yield. This results in computational effort savings and a good response for online tasks. The layout sensitivity is defined as the ratio of the area where a defect must be situated to introduce a fault to the total layout area.

$$S(x) = \frac{A_{critical}}{A_{layout}}$$

A sensitivity analysis reveals the endurance of the mask for different defect sizes, and it shows in fact the cummulative probability that a mask will fail due to defect sizes less or equal to a predefined size. However, the probability of occurrence of each defect size is different. Furthermore, the probability of occurrence of very large defects is almost neglectable. Therefore the sensitivity analysis cannot reflect what would happen to the layout in a real manufacturing environment whereas a layout probability of failure analysis does. This layout probability of failure results from the probability that the layout will fail due to a defect of a certain size and the probability that the defect will in fact occur in the manufacturing conditions. For this reason it is necessary to characterize a defect size distribution. Several analytical defect size distributions [1] [7] [9] that can do this have appear in the literature. We adopted the one presented in [7] due to its flexibility of being adapted to real manufacturing conditions. The distribution is expressed as:

$$D(x) = D_q(x) = \frac{Kx^q}{x_{peak}^{q+1}} \qquad 0 \le x \le x_{peak}$$

$$D(x) = D\ p_i\ (x) = \frac{Kx_{peak}^{p-1}}{x} p_i \qquad x_{peak} \le x \le x_{\max}$$

$$D(x) = 0 \qquad x_{\max} < x$$

$$K = \frac{(q+1)(p-1)}{(q+p) - (q+1)\left(\frac{x_{peak}}{x_{\max}}\right)^{p-1}} \qquad p_i \neq 1$$

$$K = \frac{(q+1)}{1 + (q+1) ln\left(\frac{x_{\max}}{x_{peak}}\right)} \qquad p_i = 1$$

We can express now the average POF as

$$\Phi = \int_{\min}^{\max} D(x)\, S(x)\, dx$$

where min and max are the minimum and maximum defect sizes, respectively. The POF shows if given the defect size distribution only one defect per defect size occurs. In manufacturing environments this is unreal. A density of defects of all the sizes exists. Therefore in order to predict the layout yield it is necessary to take into account this defect density. The number of faults is directly related to the number

of defects that can appear. This can be expressed as

$\lambda = \Phi * defect_density * A_{layout}$

where Φ is the average probability of failure and A_{layout} is the area of the IC.

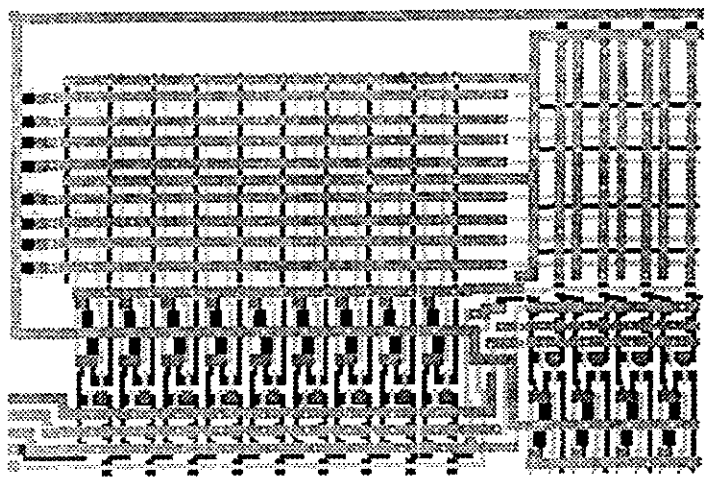
The layout yield is the probability of manufacturing the artwork without faults. To compute yield it is necessary to consider the spatial distribution of defects. Defect densities vary from wafer to wafer [18] , from region to region within the wafer, and even somtetimes tend to cluster in several wafer sections. An accurate model that takes all these factors into account is presented in [14] and is the one we incorporated in our system. The yield model is presented as

$$Y = (1 + \beta \lambda)^{\frac{-1}{\beta}}$$

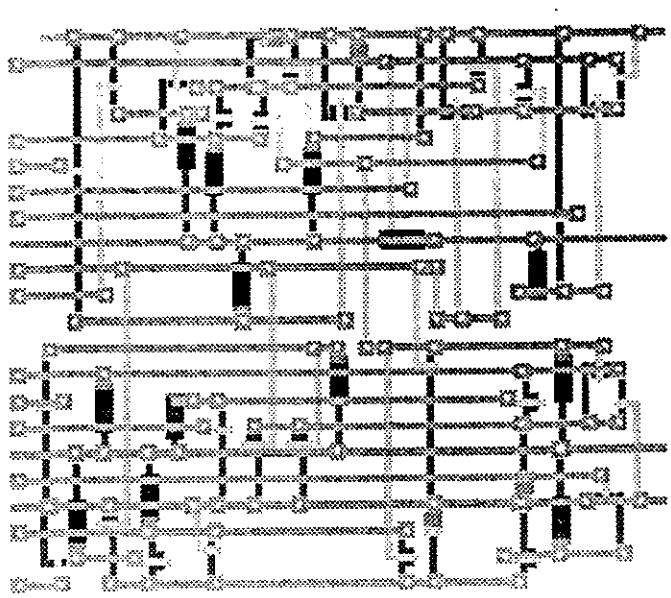
where β is the defect density clustering parameter.

7. LAYOUT ANALYSIS.

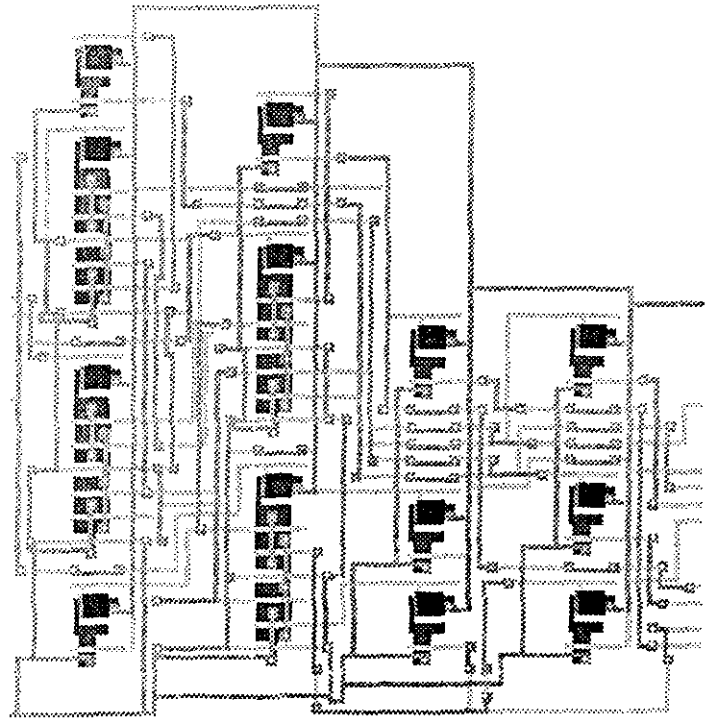
We implemented a combinational function in three different layout styles, namely a Programmable Logic Array (PLA), a Transistor Matrix (TM) [15], and a Standard Cells place and route approach (STD) [16], see Fig. 14. The layouts are for an NMOS technology of 6μm of minimum resolution features. Table 1 shows the total area and dimensions of each layout in (μm units).



(a)



(b)



(c)

metal poly diffusion

Figure 14. (a)PLA. (b)TM. (c)STD

To give the reader a feeling of how the critical regions are displayed we show in Fig. 15 the poly layer of the TM and its critical regions for bridges and cuts for defect sizes of 30μm and 12μm respectively.

We analysed each layout for defects in the range from 1 to 100μm. The sensitivities for bridges and cuts of each mask are shown in Figs. 16 and 17 respectively.

TABLE 1. Layout dimensions

Layout	Vertical Dim.	Horizontal Dim.	Area
PLA	767	503	385801
TM	453	390	176670
STD	771	756	582876

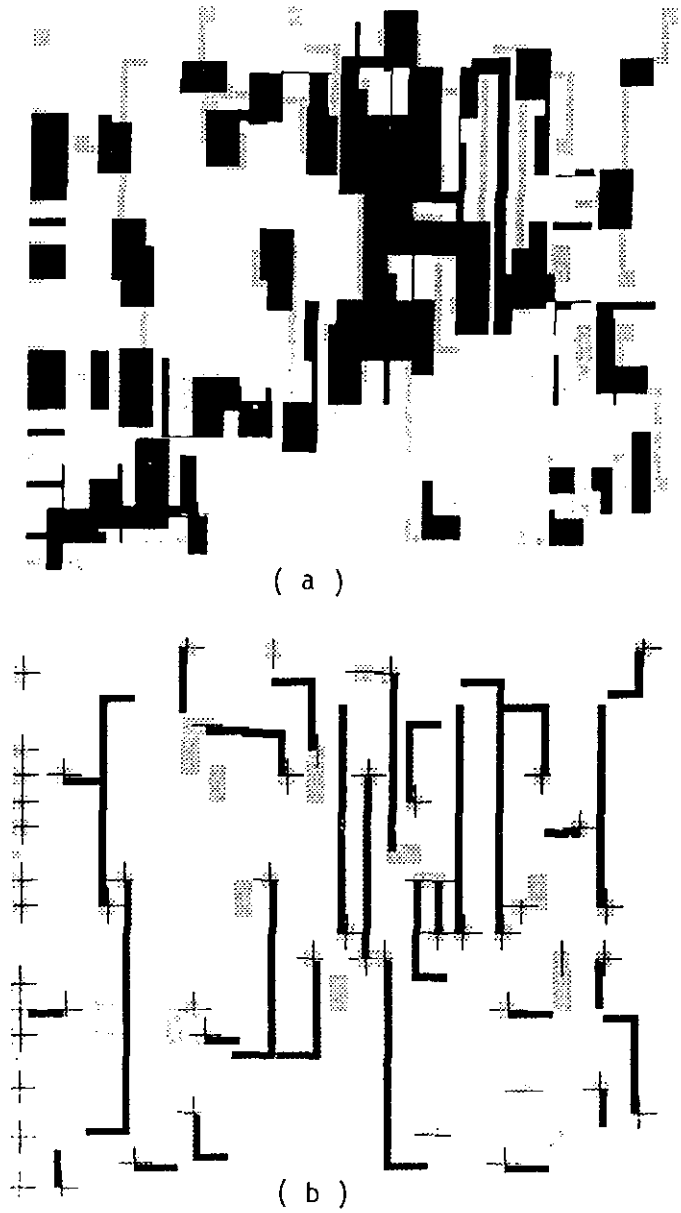


Figure 15. Critical regions are shown in black.(a)Bridges (b)Cuts.

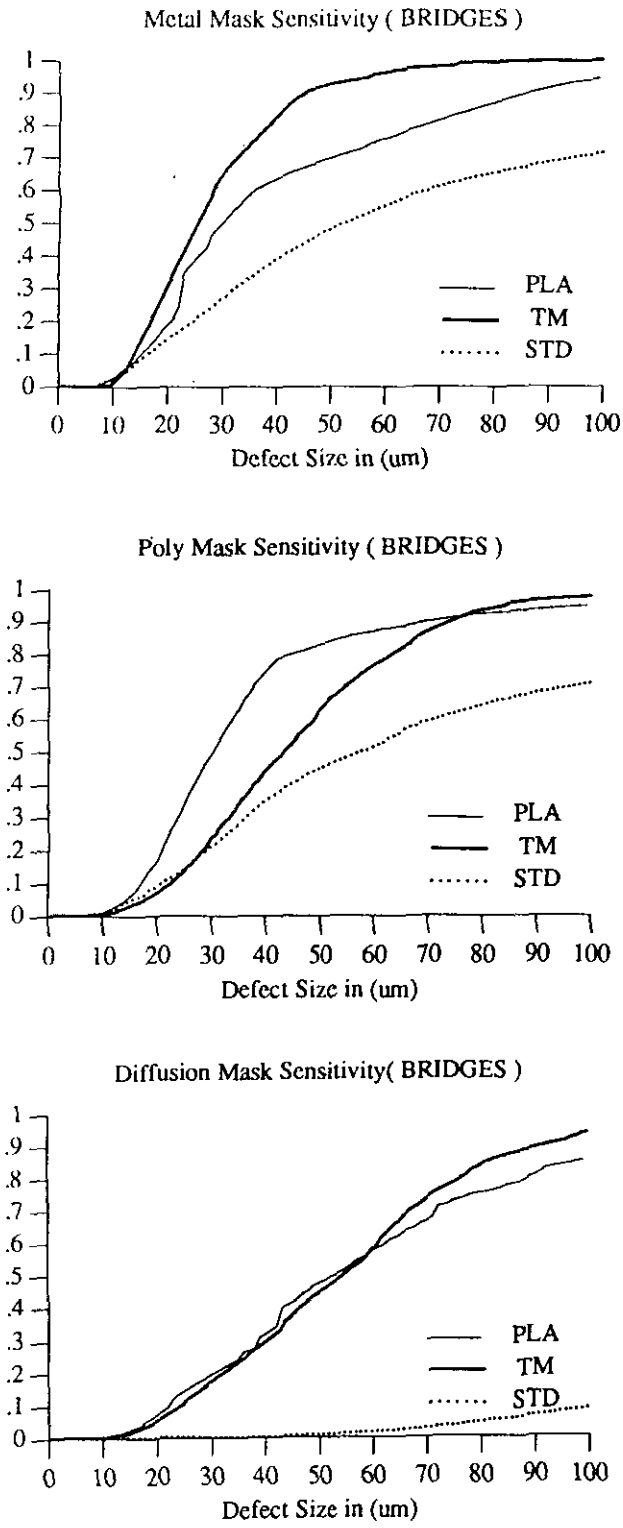


Figure 16. Layout Sensitivity for bridges

From the figures we can observe the following:

+ **Metal Mask. (BRIDGES)** The TM is the best compromise for small defects in the range from 6 to more or less 15µm. This is mainly because it is the smallest layout and thus the total critical area is

less than the other ones. However as the defect size increases, the sensitivity raises with a steep slope due to the fact that the metal mask is laid out in a very regular manner, and because the spacing between lines is small. We can see that the PLA follows the same trend but with a less steep slope, mainly due to a relaxed spacing between wires and also due to a less number of adjacent lines which reduces the chances of a bridge among them. On the other hand the STD is the best compromise for very large defects. The layout style is very relaxed, the four channels of metal wires are laid out very apart from each other, and also notice the right top corner of the layout where there is a big unutilized space.

+ **Poly Mask (BRIDGES).** Once more the TM is the best for small defects ranging from 6 to 25μm. We can also see that even for large defects the mask is quite tolerant. The reason is because it appears as a non-regular mask, wires are used mainly to form transistors and are used as interconnectors only when it is necessary. The PLA shows a very regular patron and thus the sensitivity raises with a steep slope. Notice that its sensitivity is almost twice as much compared to TM. The crossing in the curves is because the TM's area is smaller. The STD exhibits a quasi regular patron especially in the feed-throughs. This aspect is worthwhile noticing because one might think that because of the large empty spaces it is more tolerant to small defects.

+ **Diffusion Mask (BRIDGES).** The STD layout is in a column style. Almost all the patterns are joined together and the space between columns is big. This is why the sensitivity is extremely low. On the other hand, the style for the PLA is very regular but with a large distance between lines. The TM is not regular however the compactness of the layout puts the lines closer. It is interesting to notice these two aspects since the two effects seem to be equivalent from the sensitivity point of view. The PLA appears to have a higher sensitivity because of the input and output buffers which are not included in the TM. The slope is very slanted because even when the lines exhibit regularity they are very interconnected, i.e. many U shapes, hence catastrophic bridges are less likely to happen.

+ **Metal Mask (CUTS).** In this case the PLA is the best compromise because its lines are wider than in the other two layouts. Notice the crossing in the curves of the PLA and the STD. This is because the channel's lines in the STD are very separated which reduces the chances of a defect cutting more than one line at the same time, whereas in the PLA even when the lines are wider they are closer to each other. The TM simply cannot tolerate large defects.

+ **Poly Mask (CUTS).** The TM is the best compromise due to the scarce appearence of the lines. The abrupt jump for defects in the range of 12 to 15μm is a reflection of many square patterns in which the critical area is twice as much, to be precise in the form of a cross as shown in Fig. 18. The PLA shows again a steep slope due to the regularity in its layout. The STD proofs to be the best for large defects due to the empty spaces.

+ **Diffusion Mask(CUTS).** In the STD approach the use of gates makes the ratio of the transistors bigger in order to compensate delays. This results in wider patterns which are less likely to be cut by small defects, a fact which is reflected in its sensitivity. The PLA's and TM's sensitivities are very similar due to the reasons explained above.

The sensitivity analysis gives an insight of the way the masks are laid out. It reveals the endurance of the masks for different defect sizes. However, the probability of occurrence of each defect size is not the same. Furthermore, the probability that a defect of a very large size occurs is almost neglectable and hence the sensitivity analysis cannot reflect what would happen to the layout in a manufacturing

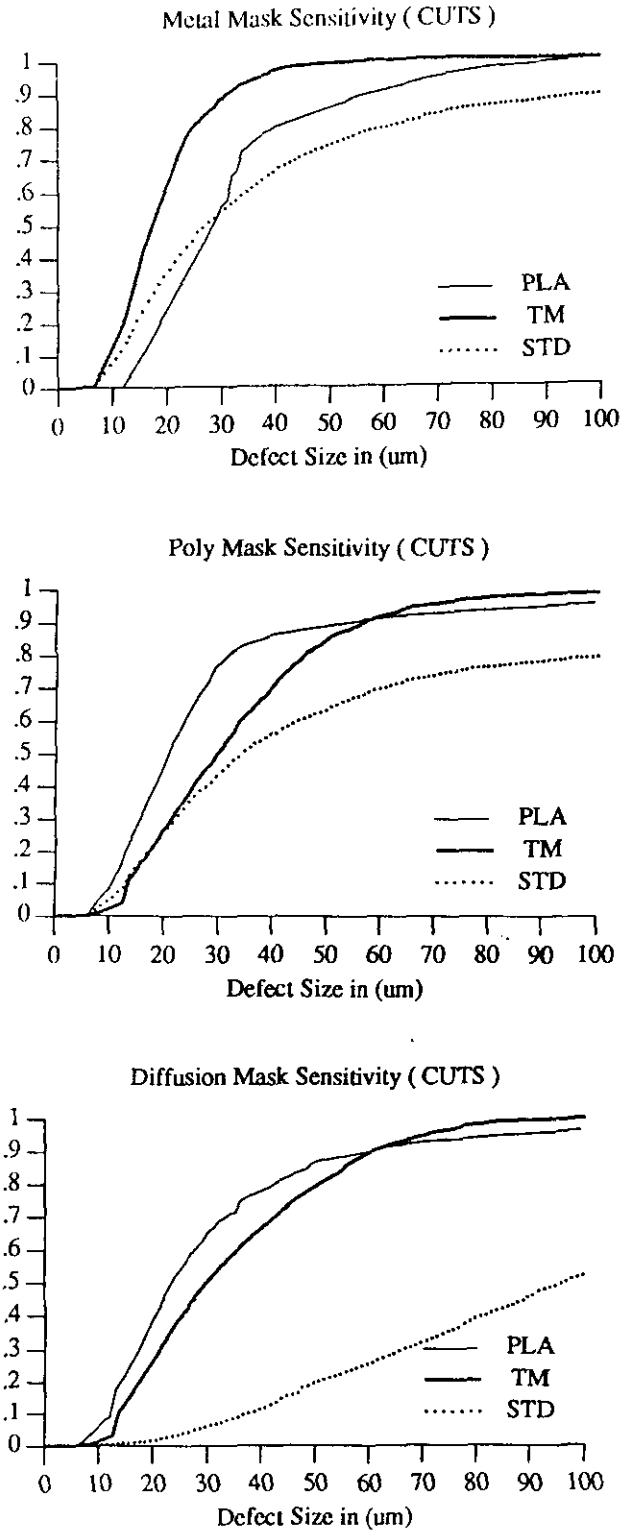


Figure 17. Layout sensitivity to cuts

environment, whereas the layout probability of failure (POF) does. We selected three different defect size distributions, one for each mask (in manufacturing lines the defect size distributions are seldom the same for all the masks), see Fig. 18. We made on purpose the distribution for the diffusion mask have a

long tail for large defect sizes, the distribution for the poly mask to peak at defects larger than the minimum resolution of 6 μm, and finally the one for the metal mask to represent a mature process.

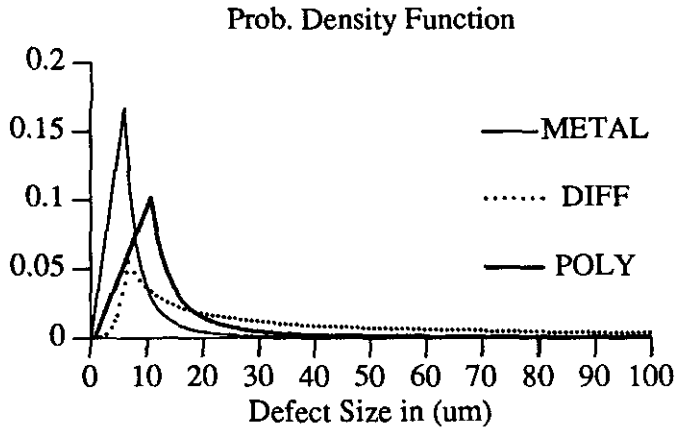


Figure 18. Defect Size Distribution

The layout probability of failure for bridges and cuts is shown in Figs. 19 and 20.

+ **Metal Mask (BRIDGES).** Something that is worthwhile noticing here is the similitude in the POFs of the PLA and STD. It is customary to think that the larger the layout the more likely to fail. In fact the curves show the contrary. The explanation to this is because the PLA uses long lines in a very uniform patron even when sometimes they are used to connect just one transistor. This style increases the risk of catching unnecessary defects along the lines. The STD uses the lines also to interconnect, however they are not in a regular style and furthermore there are many empty spaces among them. The right shift in the curve of the TM, with respect to the other two, confirms its safeness for small defects. Although for large defects it is quite unreliable.

+ **Poly Mask (BRIDGES).** A remarkable aspect is that the occurrence of defects in the size range from 15 to 30μm seems not to affect drastically the TM despite that the defect size distribution peaks at 10μm. Once more, the regularity in the PLA is its major drawback. The STD with its empty spaces is more reliable to large defect sizes.

+ **Metal Mask (CUTS).** Since the probability of having large defects is very small the PLA proofs to be the safest design with its wide lines. The STD and the TM are very similar due to the fact that their lines are narrow (6μm). The TM shows a higher POF because the lines are laid out uniformly and because of the compactness of the layout.

+ **Poly Mask (CUTS).** The scarcity in the lines of the TM and the quasi-regular lines of the STD results in two layout styles which have more or less the same probability of failure, although for small defect sizes the TM is better. The PLA has once again the highest POF.

+ **Diffusion Mask (BRIDGES & CUTS).** We can see here the dramatic differences in layout style reliabilities when they are not correctly "tuned" for a defect environment in a manufacturing line. Notice that eventhough the probability of occurence of large defects is big the STD style remains atonishing low whereas for the PLA and the TM is very high.

The layout probability of failure shows what would happen if given a defect size distribution only one defect for each defect size occurs. The next step in the analysis is to find the expected layout yield for a

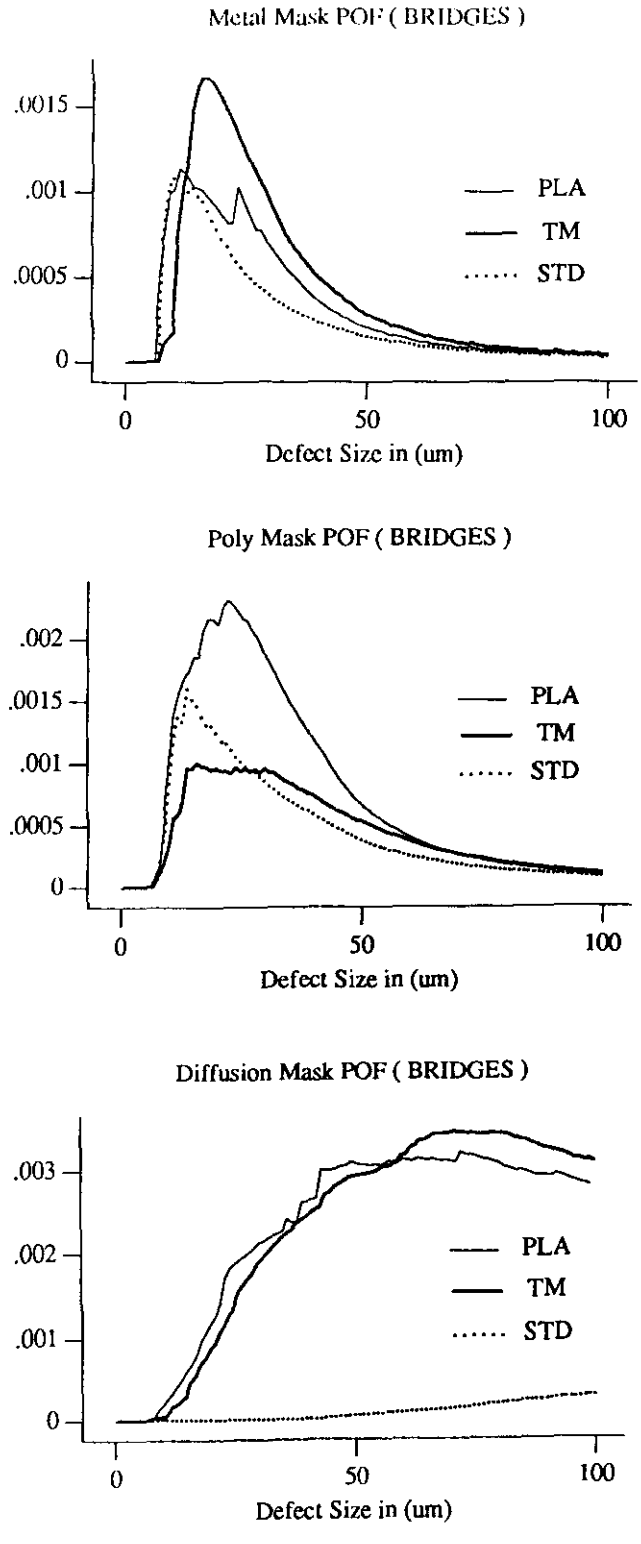


Figure 19. Layout Probability of Failure (POF) for bridges

defect density. We averaged the probability of failure for defect sizes of 1μm to 100μm and simulated the yield for defect densities varying from 1 to 10000 defects per square centimeter with β equal to 1.11. The results obtained are shown in Fig. 21 and the curves speak for themselves.

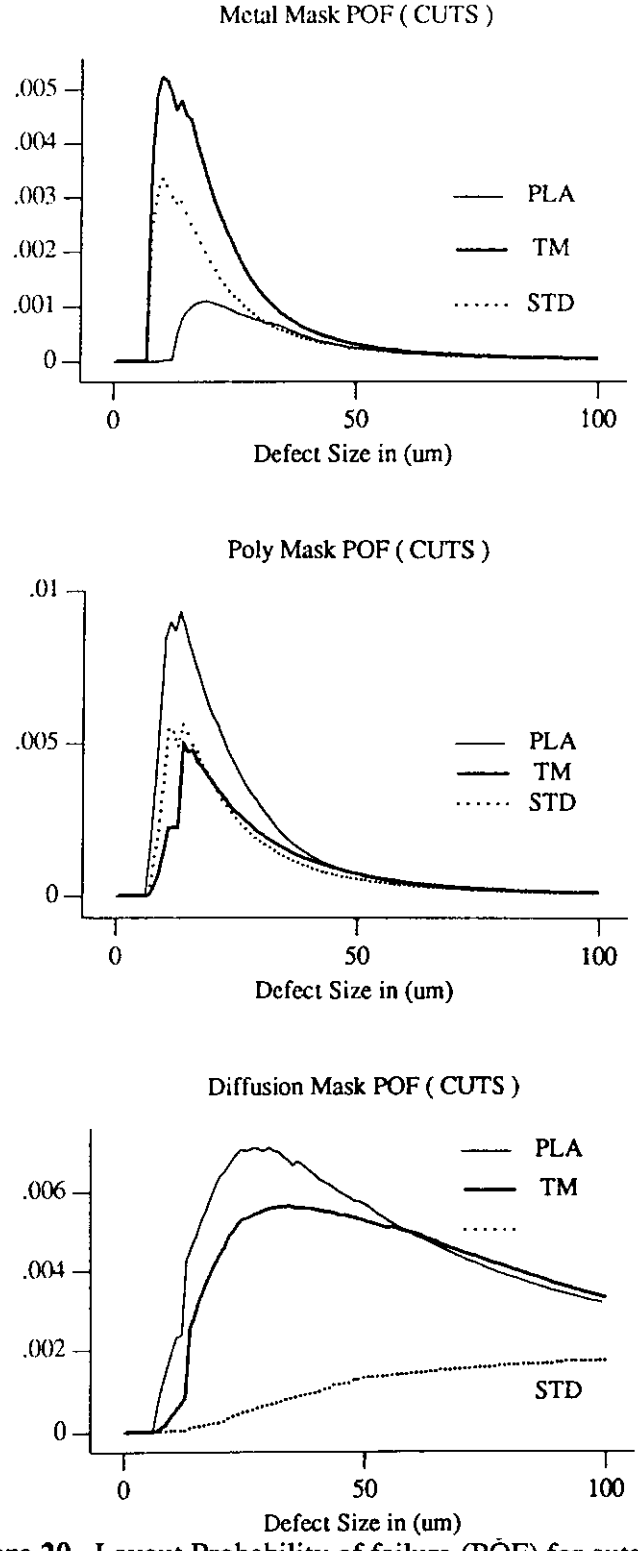
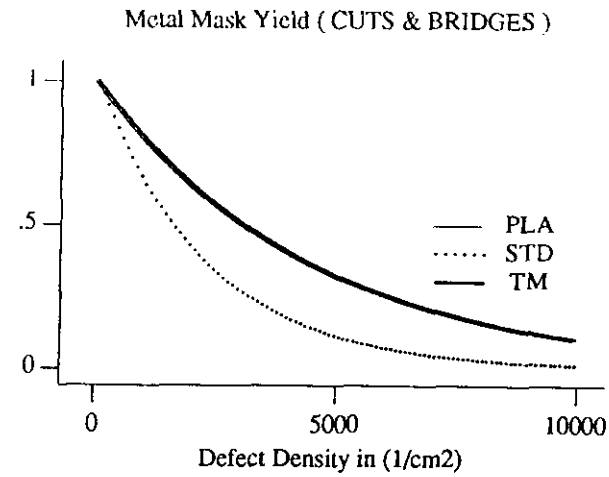
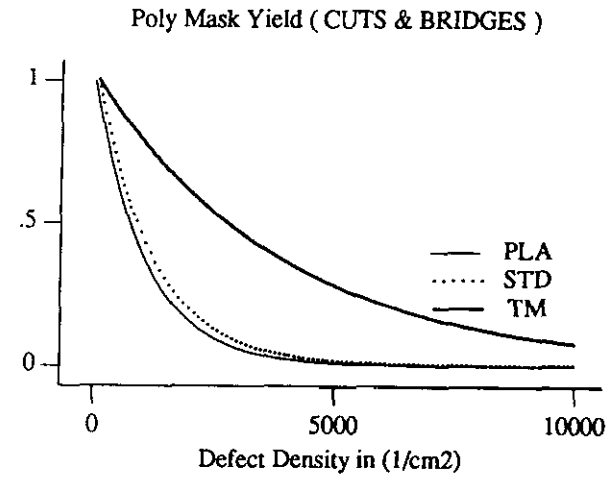


Figure 20. Layout Probability of failure (POF) for cuts

The total layout yield is presented in Fig. 22. One remarkable conclusion is that the largest layout is not always the most likely to fail, as it is depictured in the figure. On the other hand, regular layouts, like the PLA are not always very reliable. This aspect should be taken into account since PLAs are frequently used in controllers and these modules play an important role in many designs.





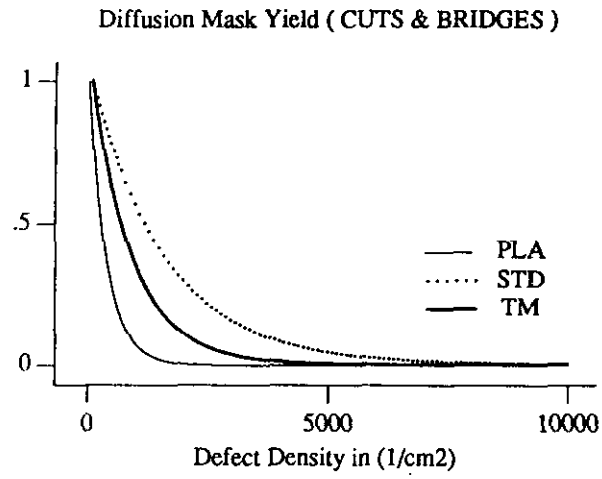


Figure 21. Mask yield

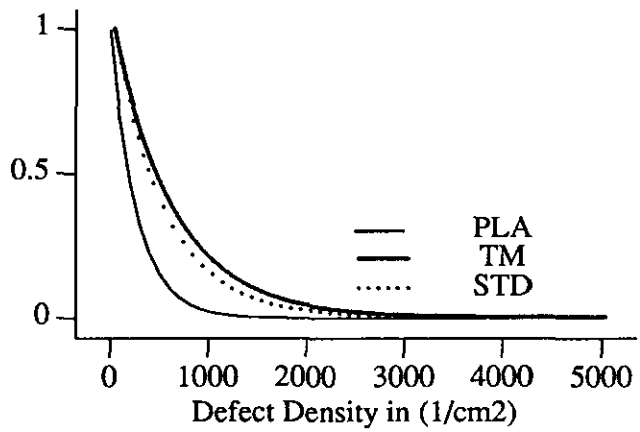


Figure 22. Final Layout yield

8. CONCLUSIONS

The significance of the critical areas is relevant to the study of layout yield. We presented a system capable of finding the critical areas in complex layouts. Unlikely to statistical Monte Carlo simulations the implementation is based purely on the geometry of the patterns. Two properties that distinguish our approach are that the critical areas for any range of defect sizes can be found by doing just one layout extraction, and that the critical areas can be displayed on the screen. These features ease the design verification task and allow to highlight the "weakest" portions of IC artwork. An exhaustive analysis of three different layout styles was carried on to compare their endurance in a manufacturing environment.

REFERENCES

[1] Stapper, C.H.
Modeling of integrated circuit defect sensitivities.
IBM J. Res. & Dev., Vol. 27(1983), p. 549-557.

[2] Stapper, C.H.
Modeling of defects in integrated circuit photolithographic patterns.
IBM J. Res. & Dev., Vol. 28(1984), p. 461-475.

[3] Ferris-Prabhu, A.V.
Modeling the critical area in yield forecasts.
IEEE J. Solid-State Circuits, Vol. SC-20(1985), p. 874-878.

[4] Shen, J.P. and W. Maly, F.J. Ferguson
Inductive fault analysis of MOS integrated circuits.
IEEE Des. & Test Comput., Vol. 2, No. 6(Dec. 1985), p. 13-26.

[5] Walker, H. and S.W. Director
VLASIC: A catastrophic fault yield simulator for integrated circuits.
IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst., Vol. CAD-5(1986), p. 541-556.

[6] Pineda de Gyvez, J. and J.A.G. Jess
On the definition of critical areas for IC photolithographic spot devices.
Paper at 1st European Test Conf., Paris, 12-14 April 1989.
Organized by: Société des Electriciens et des Electroniciens, 48, rue de la Procession, F-75724 Paris Cedex 15.

[7] Maly, W.
Modeling of lithography related yield losses for CAD of VLSI circuits.
IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst., Vol. CAD-4(1985), p. 166-177.

[8] Chen, I. and A.J. Strojwas
Realistic yield simulation for IC structural failures.
In: Digest of Technical Papers 4th IEEE Int. Conf. on Computer-Aided Design (ICCAD-86), Santa Clara, Cal., 11-13 Nov. 1986. New York: IEEE, 1986. P. 220-223.

[9] Ferris-Prabhu, A.V.
Role of defect size distributions in yield modeling.
IEEE Trans. Electron Devices, Vol. ED-32(1985), p. 1727-1736.

[10] Bentley, J.L. and T.A. Ottmann
Algorithms for reporting and counting geometric intersections.
IEEE Trans. Comput., Vol. C-28(1979), p. 643-647.

[11] Preparata, F.P. and M.I. Shamos
Computational geometry: An introduction.
Berlin: Springer, 1985.
Texts and monographs in computer science

[12] Maly, W. and J. Deszczka
Yield estimation model for VLSI artwork evaluation.
Electron. Lett., Vol. 19(1983), p. 226-227.

[13] Maly, W.
Realistic fault modeling for VLSI testing.
In: Proc. 24th IEEE/ACM Design Automation Conf., Miami Beach, Fla., 28 June-1 July 1987. New York: IEEE, 1987. P. 173-180.

[14] Stapper, Jr., C.H.
On a composite model to the IC yield problem.
IEEE J. Solid-State Circuits, Vol. SC-10(1975), p. 537-539.

[15] Ginneken, L.P.P.P. van and J.T.J. van Eijndhoven, J.A.H.C.M. Brouwers
Doubly foulded transistor matrix layout.
In: Digest of Technical Papers 6th IEEE Int. Conf. on Computer-Aided Design (ICCAD-88), Santa Clara, Cal., 7-10 Nov. 1988. New York: IEEE, 1988. P. 134-137.

[16] Theeuwen, J.F.M. and M.R.C.M. Berkelaar
Logic optimisation with technology and delay in mind.
Paper 3.3. in: Proc. Int. Workshop on Logic Synthesis, Research Triangle Park, N.C., 12-15 May 1987.
Available from: Microelectronics Center of North Carolina, P.O. Box 12889, Research Triangle Park, NC 27709, U.S.A.

[17] Maly, W. and A.J. Strojwas, S.W. Director
VLSI yield prediction and estimation: A unified framework.
IEEE Trans. Comput.-Aided Des. Integrated Circuits & Syst., Vol. CAD-5(1986), p. 114-130.

[18] Stapper, C.H.
The effects of wafer to wafer defect density variations on integrated circuit defect and fault distributions.
IBM J. Res. & Dev., Vol. 29(1985), p. 87-97.

LASER

User's Manual

CONTENTS

1. USER's MANUAL. 1
1.1 USER INTERFACE. 1
2. MAIN OPTIONS. 3
3. LAYOUT. 4
4. MASK. 5
5. VIEW. 6
6. ANALYSIS 7
7. DIAGNOSTICS AND TROUBLESHOOTING 8
7.1 USER DIAGNOSTICS 8
7.2 SYSTEM DIAGNOSTICS 8

LIST OF FIGURES

Figure 1. Main options of the system 3
Figure 2. Selection of layouts 4
Figure 3. Selection of masks 5
Figure 4. View menu 6
Figure 5. Options for analysis 7

1. USER's MANUAL.

LASER (LAyout Sensitivity ExploreR) is a system for IC layout yield prediction due to catastrophic spot defects. The system finds interactively the critical regions susceptible to local deformations, and computes also the layout sensitivity. The user can adjust the defect size distribution (according to the manufacturing environments) in order to find the probability of failure of the artwork. Once that the POF is found the yield of the artwork for a span of defect densisities (in number of defects per cm2) can be computed. Defect clustering can also be taken into account.

1.1 USER INTERFACE.

LASER assumes that the current directory contains the layouts to be analysed. Each mask in the layout is treated independently of the others and defects are modelled as squares.

— **MOUSE INTERFACE.** LASER is a highly interactive system that makes use of mouse based interface systems. The left button of the mouse is used to point at any of the options in the menus. If the system is in "critical mode" it increments the defect size and the critical regions are displayed onscreen. In this mode the middle button is used to decrement the defect size. The right most button is used to exit the system by clicking it twice.

— **INPUT FORMAT.** The input to LASER consists of files describing the layout masks. The layouts are for a Manhattan type of geometry and the layouts have to be flat, i.e. no hierarchy is allowed. The format is as follows.

<mask_name> x1 x2 y1 y2

where

mask_name is the name of the mask

x1,x2,y1,y2 are the coordinates of the mask pattern

The file's name must be

<file_name>.crt

— **ERROR REPORT.** If there were errors in the input file, LASER generates a file containing information of these errors, i.e. that x2 was less than x1, etc. The name of this file is

laser.error

Mistakes committed during the session are reported in the "interface line" (at the bottom of the screen), examples of these mistakes are that a mask was attempted to be selected without first selecting the layout.

— **INVOCATION.** LASER has a configure file that presets:

+ the minimum defect size
+ defect size step
+ colors of each one of the masks

The name of this file is laser.tech and the syntax is as follows:

min_size = CONSTANT

step_size = CONSTANT

masks = type | masks

type =(R,G,B) mask_name
where R,G,B, identify the maximum amount of red (R), green (G), blue (B). This file has to exist prior to invoking LASER. LASER assumes that the technology file is in the current directory if it is not specified in the invocation. To start LASER type critical [technology_file] <CR>

— **SCREEN ORGANIZATION.** The "menu area", to the left, is used to select any of the menu's options. The "interface line", at the bottom is used by LASER to display messages and to capture input data for the analysis. The "display area" is the section where the layout is displayed and also where the results of any of the analysis performed are projected.

2. MAIN OPTIONS.

This is the main menu of the system. From here it is possible to select a layout and the mask to be analysed, to carry on an analysis, to put the system in "critical mode", to make viewing operations, like zoom, and to hardcopy the "display area".

LASER works in two modes, "passive mode" which allows to view the layout, and "critical mode" that allows to find the catastrophic regions due to different defect sizes.

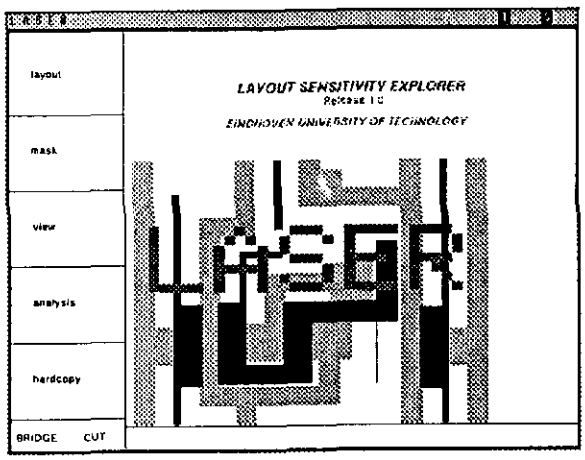


Figure 1. Main options of the system

— **LAYOUT.** This option allows to select any new layout for analysis. Every time that a new layout is selected the system is reset to "passive mode" and any analysis previously computed is lost.

— **MASK.** This option allows to select any of the masks of the previously selected layout. Every time that a new mask is selected the system is reset to passive mode.

— **VIEW.** This allows to do zoomming operations on the layout.

— **HARDCOPY.** This option is used to make a hardcopy of the display area. LASER creates a postscript file called "laser.print" that the user can spool to the printer.

— **ANALYSIS.** This option allows to analyse the mask and its local deformations such as sensitivity, probability of failure and yield.

— **BRIDGE or CUT.** These items put the system in "critical mode". Before changing the system's mode the layout mask has to be selected. When the mask is selected for the first time a message "Processing susceptible sites" appears to indicate that LASER is collecting all the places where faults can occur. When the system is ready the critical regions can be displayed on screen by clicking the left button. The critical regions are displayed in white and any intersections with the original mask are shown in cyan. Every critical region displayed corresponds to the defect size in turn. The defect size can be incremented by clicking the left button of the mouse, and decremented by clicking the middle button. The defect size increment is equal to the step size indicated in the technology file.

3. LAYOUT.

This option selects any of the layouts existing in the current directory. To pick up one layout, move the cursor to its name and click the left button of the mouse. A message "Reading layout" appears to indicated that the system is constructing the layout. If there are more layouts than the ones that are displayed move the cursor to the I option and click the left button of the mouse. To go back to the previous frame click in the I option. To go to the main menu click the left button anywhere in the "display area".

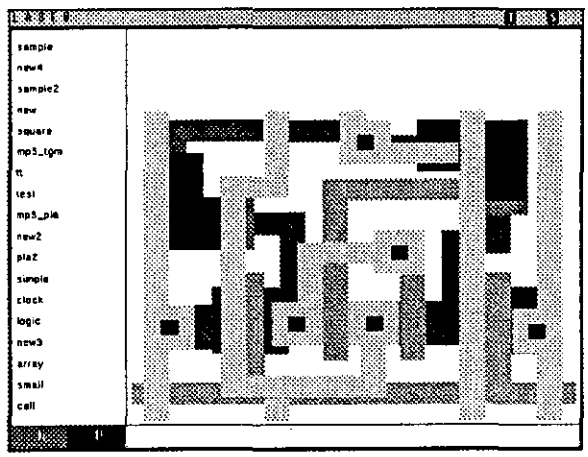


Figure 2. Selection of layouts

4. MASK.

This option selects any of the masks existing in the current layout. To pick up one mask, move the cursor to its name and click the left button of the mouse. If there are more masks than the ones that are displayed move the cursor to the ! option and click the left button of the mouse. To go back to the previous frame of masks click in the ! option. To go to the main menu click the left button anywhere in the "display area".

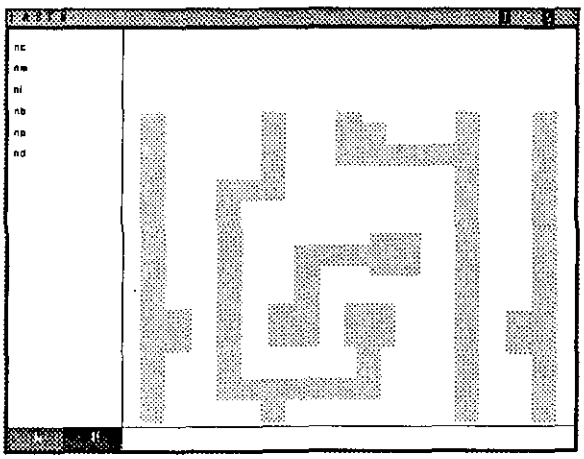


Figure 3. Selection of masks

5. VIEW.

This menu allows to zoom-in, and zoom-out the layout (mask). Also if it is not necessary to analyse the whole layout it can be bounded in this menu. If the layout is not bounded, when going back to the main menu the display area will show again the original layout (or mask).

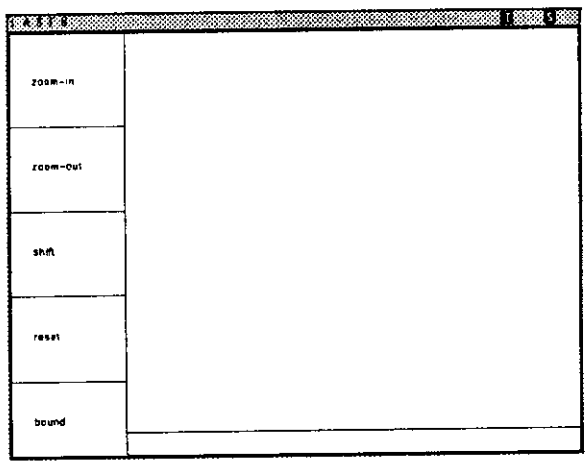


Figure 4. View menu

— **ZOOM-IN** This option allows to zoom-in the layout (or mask). A cross at the top left corner of the screen appears. Move the cross to the layout and click the left button. A rubberbox appears. Extend the box up to where the zoom operation is desired and then click the left button. LASER will show the section of the layout enclosed in the rubberbox in its entire display area.

— **ZOOM-OUT** This option allows to zoom-out the layout (or mask). A cross at the top left corner of the screen appears. Move the cross to the layout and click the left button. A rubberbox appears. LASER calculates a demagnifying factor equal to the extensions of the layout boundaries divided by the extension of the rubberbox. The display area shows the layout demagnified.

— **SHIFT** This option allows to move the layout from its original position in the display area. A cross appears at the top left corner of the screen. The shifting is calculated by how much is the cross displaced from the center of the layout. If the cross is placed to the right of the layout, the layout is moved to the right. The same applied for left, up and down.

— **RESET** If any zoom or shift operations were performed, by clicking in this option the layout or mask that was initially active before entering this menu is redisplayed.

— **BOUND** This option allows to bound the layout to any section previously zoomed-in. Only layouts can be bounded, masks not. A message "Bounding layout" appears to indicate that the operation is carried on.

6. ANALYSIS

This menu allows to analyse the sensitivity, probability of failure and the yield of the layout. Any analysis will be performed up to the current defect size (obtained by clicking the left or right mouse buttons) when the system was in "critical mode". To select any of the options of this menu move the cursor to the option area and then click the left button. To go back to the previous menu click the left button anywhere in the "display area". When LASER asks for data only the numbers, CR, Backspace, "." and the "-" sign are enabled. To indicate to LASER that the data is ready type <CR>.

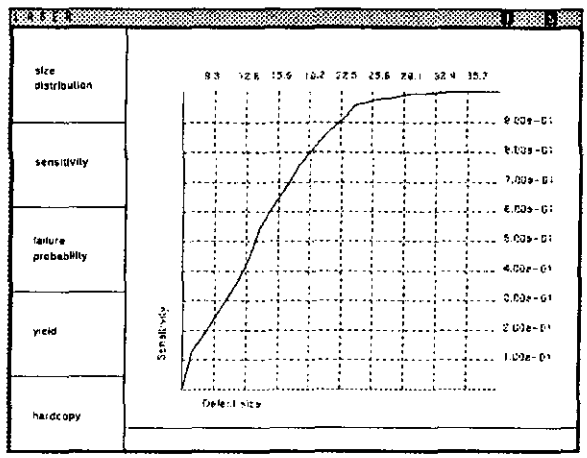


Figure 5. Options for analysis

— **SIZE DISTRIBUTION** This option allows to set the defect size distribution according to three parameters: 1) the defect peak size, 2) "p" the monotonously incrementing part of the distribution, and 3) "q" the monotonously decrementing part of the distribution. For details of this distribution refer to the technical section 6 of the report. LASER asks one parameter at the time in the interface line. The data is also captured in this line. The distribution shown is the pdf of defects.

— **SENSITIVITY.** LASER computes the sensitivity of which either mode BRIDGE or CUT was active.

— **FAILURE PROBABILITY.** Once that the defect size distribution and the sensitivity are computed the probability of failure can be found. If any of the sensitivity or defect size distribution is not previously computed LASER will report so.

— **YIELD** In order to calculate the yield the sensitivity, defect size distribution and probability of failure have to be computed. The yield is calculated according to three parameters: 1)minimum defect density, 2) maximum defect density, and 3)defect cluster parameter. The densities have to be given in number of defects per cm2.

7. DIAGNOSTICS AND TROUBLESHOOTING

There are basically two types of diagnostics. One is referred as user diagnostics and it concerns all the interface messages between the user and LASER. The other types of diagnostics are referred as system diagnostics. These messages deal with the internal programming of LASER such as to allocate memory space for layout structures, or to search for layout files, etc. The latter type of diagnostics appears only when there are conflicts between LASER and the operating system.

7.1 USER DIAGNOSTICS

The following are the error messages displayed by LASER. A brief explanation of its meaning is given. These messages appear as a result of an incorrect usage of the features of LASER. Their interpretation during the session is straightforward.

Layout file is corrupted, check 'laser.errors' for errors ...
: The input data of the layout file contained errors.

Select first the mask
: An attempt to put the system in critical mode was done. The system is in critical mode only when the mask is known

Layout has not been selected
: An attempt to pick up a mask without previously selecting the layout.

Mode has not been selected
: An attempt to display the critical regions was made. Select first the mode: BRIDGES or CUTS.

Nothing to be hardcopied
: No layout has been selected to be analyzed, thus there is nothing to be copied.

Only entire layouts can be bounded
: An attempt to bound a single mask was done.

Defect size is undefined
: An attempt to find the defect size distribution was done. Find first the critical regions up to the defect size for what you want to compute the defect size ditribution.

Critical regions have not been formed
: The system cannot compute the sensitivity without that the user finds the critical regions first.

Sensitivity undefined
: Compute first the sensitivity and then find the probability of failure.

Probability of failure is undefined
: Compute first the probability of failure before trying to compute the yield.

7.2 SYSTEM DIAGNOSTICS

The following are the messages that can appear as a consequence of a conflict between LASER and the operating system. They appear as

[SYSXX]Message

where XX is the number of the message, and Message is a short legend of the problem. If any of these messages appears
1. check ownership of directories
2. check that the directories are accesible
3. check that the files are readable
4. check that the files are writable
5. check that there is enough memory usage space
6. consult with your system manager for help

The diagnostics are:

1. Cannot allocate space to plot the defect size.
2. Cannot open file "laser.print".
3. Cannot allocate space to plot the defect density.
4. Cannot allocate space to create the layout.
5. Cannot open the layout file.
6. Cannot allocate space to save the critical regions for bridges.
7. Cannot allocate space to save the critical regions for cuts.
8. Cannot reallocate space to save the critical regions for bridges.
9. Cannot reallocate space to save the critical regions for cuts.
10. Cannot allocate space to save the critical areas for bridges.
11. Cannot allocate space to save the critical areas for cuts.
12. Cannot allocate space to save the layout probability of failure (POF).
13. Cannot allocate space to save the defect size for the layout POF.
14. Cannot initialize data structure (binary trees).
15. Cannot insert member in the binary tree.
16. Cannot allocate space to save the sensitivity to bridges.
17. Cannot allocate space to save the sensitivity to cuts.
18. Cannot allocate space to save the defect size for the defect size distr.
19. Cannot allocate space to save the defect size distribution.
20. Cannot allocate space to create susceptible sites.
21. Cannot allocate space to initialize the colors of the layers
22. Cannot allocate space to display the set of layers on screen.
23. Cannot allocate space to save the layers found in the tech file.
24. Cannot allocate space to perform the horizontal sweep of the layout
25. Cannot allocate space to perform the vertical sweep of the layout
26. Cannot allocate space to save the defect density.
27. Cannot allocate space to save the yield

Eindhoven University of Technology Research Reports
Faculty of Electrical Engineering

ISSN 0167-9708
Coden: TEUEDE

(188) Jóźwiak, J.
THE FULL DECOMPOSITION OF SEQUENTIAL MACHINES WITH THE STATE AND OUTPUT BEHAVIOUR REALIZATION.
EUT Report 88-E-188. 1988. ISBN 90-6144-188-9

(189) Pineda de Gyvez, J.
ALWAYS: A system for wafer yield analysis.
EUT Report 88-E-189. 1988. ISBN 90-6144-189-7

(190) Siuzdak, J.
OPTICAL COUPLERS FOR COHERENT OPTICAL PHASE DIVERSITY SYSTEMS.
EUT Report 88-E-190. 1988. ISBN 90-6144-190-0

(191) Bastiaans, M.J.
LOCAL-FREQUENCY DESCRIPTION OF OPTICAL SIGNALS AND SYSTEMS.
EUT Report 88-E-191. 1988. ISBN 90-6144-191-9

(192) Worm, S.C.J.
A MULTI-FREQUENCY ANTENNA SYSTEM FOR PROPAGATION EXPERIMENTS WITH THE OLYMPUS SATELLITE.
EUT Report 88-E-192. 1988. ISBN 90-6144-192-7

(193) Kersten, W.F.J. and C.A.P. Jacobs
ANALOG AND DIGITAL SIMULATION OF LINE-ENERGIZING OVERVOLTAGES AND COMPARISON WITH MEASUREMENTS IN A 400 kV NETWORK.
EUT Report 88-E-193. 1988. ISBN 90-6144-193-5

(194) Hosselet, L.M.L.F.
MARTINUS VAN MARUM: A Dutch scientist in a revolutionary time.
EUT Report 88-E-194. 1988. ISBN 90-6144-194-3

(195) Bondarev, V.N.
ON SYSTEM IDENTIFICATION USING PULSE-FREQUENCY MODULATED SIGNALS.
EUT Report 88-E-195. 1988. ISBN 90-6144-195-1

(196) Liu Wen-Jiang, Zhu Yu-Cai and Cai Da-Wei
MODEL BUILDING FOR AN INGOT HEATING PROCESS: Physical modelling approach and identification approach.
EUT Report 88-E-196. 1988. ISBN 90-6144-196-X

(197) Liu Wen-Jiang and Ye Dau-Hua
A NEW METHOD FOR DYNAMIC HUNTING EXTREMUM CONTROL, BASED ON COMPARISON OF MEASURED AND ESTIMATED VALUE.
EUT Report 88-E-197. 1988. ISBN 90-6144-197-8

(198) Liu Wen-Jiang
AN EXTREMUM HUNTING METHOD USING PSEUDO RANDOM BINARY SIGNAL.
EUT Report 88-E-198. 1988. ISBN 90-6144-198-6

(199) Jóźwiak, L.
THE FULL DECOMPOSITION OF SEQUENTIAL MACHINES WITH THE OUTPUT BEHAVIOUR REALIZATION.
EUT Report 88-E-199. 1988. ISBN 90-6144-199-4

(200) Huis in 't Veld, R.J.
A FORMALISM TO DESCRIBE CONCURRENT NON-DETERMINISTIC SYSTEMS AND AN APPLICATION OF IT BY ANALYSING SYSTEMS FOR DANGER OF DEADLOCK.
EUT Report 88-E-200. 1988. ISBN 90-6144-200-1

(201) Woudenberg, H. van and R. van den Born
HARDWARE SYNTHESIS WITH THE AID OF DYNAMIC PROGRAMMING.
EUT Report 88-E-201. 1988. ISBN 90-6144-201-X

(202) Engelshoven, R.J. van and R. van den Born
COST CALCULATION FOR INCREMENTAL HARDWARE SYNTHESIS.
EUT Report 88-E-202. 1988. ISBN 90-6144-202-8

(203) Delissen, J.C.M.
THE LINEAR REGRESSION MODEL: Model structure selection and biased estimators.
EUT Report 88-E-203. 1988. ISBN 90-6144-203-6

(204) Kalasek, V.K.I.
COMPARISON OF AN ANALYTICAL STUDY AND EMTP IMPLEMENTATION OF COMPLICATED THREE-PHASE SCHEMES FOR REACTOR INTERRUPTION.
EUT Report 88-E-204. 1988. ISBN 90-6144-204-4

Eindhoven University of Technology Research Reports
Faculty of Electrical Engineering

ISSN 0167-9708
Coden: TEUEDE

(205) Butterweck, H.J. and J.H.F. Ritzerfeld, M.J. Werter
FINITE WORDLENGTH EFFECTS IN DIGITAL FILTERS: A review.
EUT Report 88-E-205. 1988. ISBN 90-6144-205-2

(206) Bollen, M.H.J. and G.A.P. Jacobs
EXTENSIVE TESTING OF AN ALGORITHM FOR TRAVELLING-WAVE-BASED DIRECTIONAL DETECTION AND PHASE-SELECTION BY USING TWONFIL AND EMTP.
EUT Report 88-E-206. 1988. ISBN 90-6144-206-0

(207) Schuurman, W. and M.P.H. Weenink
STABILITY OF A TAYLOR-RELAXED CYLINDRICAL PLASMA SEPARATED FROM THE WALL BY A VACUUM LAYER.
EUT Report 88-E-207. 1988. ISBN 90-6144-207-9

(208) Lucassen, F.H.R. and H.H. van de Ven
A NOTATION CONVENTION IN RIGID ROBOT MODELLING.
EUT Report 88-E-208. 1988. ISBN 90-6144-208-7

(209) Jóźwiak, L.
MINIMAL REALIZATION OF SEQUENTIAL MACHINES: The method of maximal adjacencies.
EUT Report 88-E-209. 1988. ISBN 90-6144-209-5

(210) Lucassen, F.H.R. and H.H. van de Ven
OPTIMAL BODY FIXED COORDINATE SYSTEMS IN NEWTON/EULER MODELLING.
EUT Report 88-E-210. 1988. ISBN 90-6144-210-9

(211) Boom, A.J.J. van den
H_∞-CONTROL: An exploratory study.
EUT Report 88-E-211. 1988. ISBN 90-6144-211-7

(212) Zhu Yu-Cai
ON THE ROBUST STABILITY OF MIMO LINEAR FEEDBACK SYSTEMS.
EUT Report 88-E-212. 1988. ISBN 90-6144-212-5

(213) Zhu Yu-Cai, M.H. Driessen, A.A.H. Damen and P. Eykhoff
A NEW SCHEME FOR IDENTIFICATION AND CONTROL.
EUT Report 88-E-213. 1988. ISBN 90-6144-213-3

(214) Bollen, M.H.J. and G.A.P. Jacobs
IMPLEMENTATION OF AN ALGORITHM FOR TRAVELLING-WAVE-BASED DIRECTIONAL DETECTION.
EUT Report 89-E-214. 1989. ISBN 90-6144-214-1

(215) Hoeijmakers, M.J. en J.M. Vleeshouwers
EEN MODEL VAN DE SYNCHRONE MACHINE MET GELIJKRICHTER, GESCHIKT VOOR REGELDOELEINDEN.
EUT Report 89-E-215. 1989. ISBN 90-6144-215-X